Computers & Industrial Engineering 90 (2015) 39–45

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

# Resource-constrained scheduling with optional recycling operations

Bertrand M.T. Lin *

*Institute of Information Management, Department of Information and Finance Management, National Chiao Tung University, Hsinchu 300, Taiwan*

ARTICLE INFO

*Article history:*
Received 21 September 2014
Received in revised form 10 August 2015
Accepted 26 August 2015
Available online 31 August 2015

*Keywords:*
Relocation problem
Resource-constrained scheduling
Optional recycling operations
Complexity

ABSTRACT

This paper introduces a scheduling problem with a variant of resource constraint that stems from a relocation project. We also propose the concept of optional recycling operations, in which recycling operations are separated from the processing of jobs and recycling operations are exercised only when necessary. An integer program is proposed to formulate the problem and facilitate complexity classification. We propose a pseudo-polynomial time dynamic program, and then classify the complexity status of several restricted cases.



## 1. Introduction and problem statements

This paper introduces a scheduling problem with a new variant of resource constraints. Resource-constrained project scheduling problems (RCPSP) have been receiving considerable research attention for several decades due to their practical significance and theoretical challenges. Blazewicz et al. (1983), Brucker et al. (1999), Hartmann and Briskonn (2010), Herroelen and Leus (2005), Özdamar and Ulusoy (1995), and Weglarz et al. (2011), are excellent reviews and surveys on this subject. Resource may stand for monetary capital, computer memory, automated guided vehicle, housing unit, just to name a few. Depending on their nature and usage, resources can be characterized into several categories. A resource is called renewable if it may be reused when it is released by a task that acquired and possessed this resource. Constraints of renewable resources are cast on a period-by-period basis with an upper limit of availability through the planning horizon. Personnel, hotel rooms, and rental cars are examples among others. The counterpart category is nonrenewable resources that are constrained on a project basis. The total amount of a nonrenewable resource assigned to a project is fixed and a resource cannot be reused by or assigned to an unprocessed task if it has been assigned to any other task. A project is called doubly constrained if both renewable and nonrenewable constraints are imposed. Kolisch, Sprecher, and Drexl (1995) introduce the concept of partially (non) renewable resources referring to the availability of which is defined for only specific time intervals. Steeneck and Sarin (2015) investigate project scheduling with varying resource consumption rates of the activities. The processing lengths of the activities are defined by a time-resource trade-off function. The objective is to construct a resource-feasible schedule that attains the optimal project completion time, i.e. makespan The resource constraints addressed in this paper is a generalization of renewable and nonrenewable resources in the sense that a task when completed will release the resource it has acquired for processing subject to the flexibility allowing the amount of the resource released to be smaller than, equal to or even larger than the amount of the previously acquired resource.

The proposed problem is formally defined as follows. A set of $n$ jobs $\mathcal{N} = \{1, 2, \ldots, n\}$ is to be processed on a single machine. Each job $j \in \mathcal{N}$ is characterized by four non-negative integral parameters: (1) Resource requirement $a_j$: Amount of the resource required to commence the processing of job $j$, (2) Processing time $p_j$: Processing time of the regular operation of job $j$, (3) Resource yield $b_j$: Amount of the resource that may be returned by job $j$ when it is completed, and (4) Resource recycling time $q_j$: Time required for recycling the $b_j$ units of the resource. A common pool of $v_0$ units of a single-type resource is given for processing the jobs of $\mathcal{N}$. Job $j$ requires and consumes $a_j$ units of the resource from the resource pool to commence its processing. That is, when the job is to be processed, there must be at least $a_j$ units of the resource in the pool. The processing time is given by $p_j$. Upon its completion, we either exercise its recycling operation, taking $q_j$ units of time, or directly proceed to the processing of other jobs. If the recycling operation is carried out, then $b_j$ units of the resource will be produced and deposited into the common pool. No strict relation between $a_j$ and $b_j$ is assumed, i.e. $b_j$ can be greater than, less than, or equal

* Tel.: +886 3 5131472; fax: +886 3 5723792.
*E-mail address:* bmtlin@mail.nctu.edu.tw

http://dx.doi.org/10.1016/j.cie.2015.08.014
0360-8352/

to $a_j$. The decision consists of two parts (1) selecting a subset of recycling operations to be exercised, and (2) determining a sequence of all regular operations and the selected recycling operations. The derived sequence should be feasible subject to the resource constraint in the sense that in the course of execution no job is blocked by insufficiency of the resource. The objective is to minimize the largest completion time of the regular processing operations, i.e. the makespan. We denote the problem by $1|a_j, b_j, recl|C_{\max}$, in which the second field includes $a_j$ and $b_j$ to indicate this type of resource constraints and *recl* the recycling operations. Fig. 1 shows an instance and two feasible schedules. The two schedules exercise different sets of recycling operations and thus have different project completion times.

The studied problem is a generalization of the relocation problem that is formulated from in a redevelopment project in east Boston (Kaplan, 1986). The parameter $a_j$ refers to the original capacity of a building, and $b_j$ is the new capacity of the building after redevelopment. Having an initial budget for temporarily housing the tenants of the buildings under redevelopment, the municipal government has to determine a redevelopment sequence in which all tenants could be successfully housed. In the base definition of the relocation problem, no recycling operations are considered, i.e. a job will immediately produce and return its resource yield at the completion of its processing. Kaplan and Amir (1988) prove that the feasibility of a processing sequence subject to the given $v_0$ can be tackled using the concept of two-machine flow shop scheduling (Johnson, 1954). Note that in the base relocation problem, no temporal parameters, like processing times and due dates, are involved. Taking into account the temporal issues along the planning horizon, Kaplan (1986) addresses the reconstruction time lines of the buildings and allows multiple buildings to be processed in parallel if the available resource permits. Kononov and Lin (2006) prove that makespan minimization is strongly NP-hard even if there are only two working crews and each building takes one unit of time to redevelop. They also establish inapproximability of the problem and analyze the performance ratios of approximation algorithms. For the objective function of total weighted completion time, Kononov and Lin (2010) give strong NP-hardness proofs and analyze several approximation algorithms. The first attempt to introduce recycling operations is due to Lin and Huang (2006). A second working crew is available for performing the recycling operations. The two working crews constitute a two-machine flow shop, the first machine is for demolishing the buildings and the second machine for erecting new buildings. Their work focuses on the design of branch-and-bound algorithms and computational experiments. Cheng, Lin, and Huang (2012) study the same model but center around classification of complexity status of various special cases.

In the setting we are working on, regular operations and recycling operations are processed on a single machine rather than a two-machine flow shop as considered in Lin and Huang (2006) and Cheng et al. (2012). Moreover, not all recycling operations are required to be finished. The flexibility in choosing recycling operations is due to the supporting roles of the recycling operations which can be conducted at the back-end side after the completion of customer orders, if the resource availability allows. This new feature highlights the different roles of all operations that compete for the machine processing capacity and the combinatorial selection of recycling operations out from sequencing decisions.

The rest of this paper is organized as follows. In Section 2, we will discuss several preliminary properties of the base relocation problem and the $1|a_j, b_j, recl|C_{\max}$ problem. Section 3 is dedicated to the development of integer programs and a pseudo-polynomial time dynamic programming algorithm. Given the fact that the problem is NP-hard, complexity status of different special cases are discussed in Section 4. Concluding remarks and suggestions for further studies are given in Section 5.

## 2. Preliminaries

This section develops preliminary properties of the studied problem. We start with the introduction of known results of the base setting.

In the base relocation problem, denoted by $1|a_j, b_j|C_{\max}$, the regular and recycling operations of a job are coalesced as an aggregate unit in the sense that at the completion of the job it immediately produces and stores its resource yield to the common resource pool. For convenience in presentation, define the resource contribution of job $j \in \mathcal{N}$ as $\delta_j = b_j - a_j$. Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ denote a job sequence in the base setting. Denote by $v_j(\sigma)$ the resource level after job $\sigma_j$ produces and stores its resource yield, and before job $\sigma_{j+1}$ acquires the resource to start its processing. It is easy to see

| jobs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $a_j$ | 2 | 5 | 6 | 4 | 3 |
| $b_j$ | 5 | 11 | 12 | 2 | 1 |
| $p_j$ | 2 | 2 | 2 | 2 | 2 |
| $q_j$ | 5 | 10 | 15 | 3 | 2 |

Initial resource level $v_0 = 5$

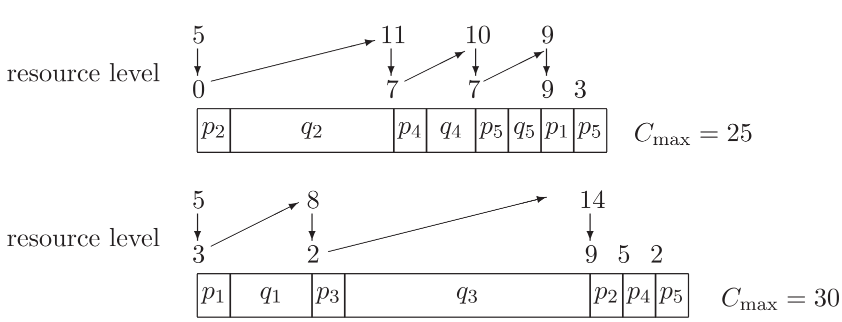


**Fig. 1.** Numerical example with two feasible schedules.